AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Previously presented) A system, comprising:

means for receiving at least one software program written in a high level language; and

means for automatically generating an instruction set architecture optimized for executing that program(s), wherein the instruction set architecture is represented as a set of configurations containing one or more extension instructions based on instructions in an existing standard or existing user defined instruction set architecture.

2. (Previously presented) The system of claim 1, wherein:

the extension instructions operate on states and register files in the existing standard or existing user-defined instruction set architecture.

3. (Original) The system of claim 2, wherein the extension instructions contain vectorized versions of the existing instructions.

4. (Original) The system of claim 2, wherein the extension instructions contain VLIW combinations of the existing instructions.

5. (Original) The system of claim 2, wherein the extension instructions contain fused combinations of the existing instructions.

6. (Original) The system of claim 2, wherein the extension instructions contain specialized versions of the existing instructions.

7. (Previously presented) The system of claim 2, wherein the extension instructions contain vectorized versions of operations supported by the high level language.

8. (Original) The system of claim 2, wherein the extension instructions contain VLIW combinations of operations supported by the high level language.

9. (Original) The system of claim 2, wherein the extension instructions contain fused combinations of operations supported by the high level language.

10. (Original) The system of claim 2, wherein the extension instructions contain specialized versions of operations supported by the high level language.

11. (Original) The system of claim 2, wherein the extension instructions contain at least two of vectorized, VLIW, fused and specialized versions of the existing instructions.

12. (Original) The system of claim 2, wherein the extension instructions contain at least two of vectorized, VLIW, fused and specialized versions of operations supported by the high level language.

13. (Canceled)

14. (Previously presented) The system of claim 2, wherein:

instruction set architecture generation is guided by analysis information gathered from the at least one software program; and

the analysis information is gathered for each region of code that could get a performance improvement from a generated instruction set algorithm.

15. (Original) The system of claim 14, wherein the analysis information includes an execution count of each region as determined from real or estimated profiling information.

16. (Original) The system of claim 14, wherein the analysis information includes an execution count of each region as determined from user-supplied directives.

17. (Original) The system of claim 14, where the analysis information includes a dependence graph of each region.

18. (Original) The system of claim 14, where the analysis information includes a set of operation vector lengths that can be used to improve performance of each region.

19. (Original) The system of claim 14, wherein each region is evaluated with a set of instruction set architecture configurations to determine a performance improvement that would result if instructions, operations, register files, and states represented by the configuration could be used for the region.

20. (Previously presented) The system of claim 19, wherein:

instruction set architecture generation uses as a guideline an estimate of hardware cost of each instruction set architecture configuration that includes a cost of logic necessary to implement instructions, operations, register files, and state represented by the configuration; and

the hardware cost and performance improvement of each instruction set architecture configuration for each region is used to determine a set of instruction set architecture configurations that together describe the generated instruction set architecture such that the performance improvement of the software program(s) is increased as much as possible while the hardware cost of the generated instruction set architecture does not exceed a cost budget.

21. (Previously presented) The system of claim 19, wherein:

instruction set architecture generation uses as a guideline an estimate of hardware cost of each instruction set architecture configuration that includes a cost of logic necessary to implement instructions, operations, register files, and state represented by the configuration; and

the hardware cost and performance improvement of each instruction set architecture configuration for each region is used to determine the set of instruction set architecture configurations that together describe the generated instruction set architecture such that the hardware cost of the generated instruction set architecture is as small as possible while providing a performance improvement that is greater or equal to a performance goal.

22. (Original) The system of claim 19, wherein the hardware cost and performance improvement of each instruction set architecture hardware configuration for each region is used to determine the set of instruction set architecture configurations that together describe the

generated instruction set architecture such that the hardware cost of the generated instruction set architecture is smaller than a predetermined function of the performance improvement.

23. (Original) The system of claim 19, wherein a performance improvement provided by a particular instruction set architecture configuration for a particular region is determined by an instruction scheduling algorithm operating on a modified dependence graph of the region.

24. (Original) The system of claim 23, wherein the dependence graph is modified to replicate operations with an operation width that is less than one.

25. (Original) The system of claim 23, wherein the dependence graph is modified to replace groups of operations with a single fused operation.

26. (Original) The system of claim 19, wherein the performance improvement provided by a particular instruction set architecture configuration for a particular region is determined using resource limits.

27. (Previously presented) The system of claim 2, wherein instruction set architecture generation uses as a guideline an estimate of hardware cost of each instruction set architecture configuration that includes a cost of logic necessary to implement instructions, operations, register files, and state represented by the configuration.

28. (Original) The system of claim 27, wherein the hardware cost is estimated by adding hardware costs of components present in the instruction set architecture configuration.

29. (Original) The system of claim 27, wherein the hardware cost is reduced to represent reduced logic necessary when specialized operations replace generic operations.

30. (**Currently Amended**) A system, comprising:

means for receiving at least one software program written in a high level language; and

means for automatically generating an instruction set architecture optimized for executing said at least one program, by adding one or more new extension instructions to instructions in an

existing **standard or existing user-defined** instruction set architecture based on the analysis of the said at least one program,

wherein the new instruction(s) contain vectorized versions of the existing instructions.

31. **(Currently Amended)** A system, comprising:

means for receiving at least one software program written in a high level language; and

means for automatically generating an instruction set architecture optimized for executing said at least one program, by adding one or more new register files based on the analysis of the said at least one program,

wherein the new register file(s) are vectorized versions of an existing standard or existing user defined ~~instruction set architecture~~ **register file(s)**.

32. (Previously presented) A system, comprising:

means for receiving at least one software program written in a high level language; and

means for automatically generating an instruction set architecture optimized for executing said at least one program, by adding one or more new extension instructions to instructions in an existing instruction set architecture based on the analysis of the said at least one program,

wherein the new instruction(s) contain specialized versions of existing standard or user-defined instructions in the existing instruction set architecture.

33. (Previously presented) A system, comprising:

means for receiving at least one software program written in a high level language; and

means for automatically generating an instruction set architecture optimized for executing said at least one program, by adding one or more new extension instructions to instructions in an existing instruction set architecture based on the analysis of the said at least one program,

wherein a single new instruction contains one of a fused, specialized and vectorized combination of existing standard or user-defined instructions in the existing instruction set architecture.